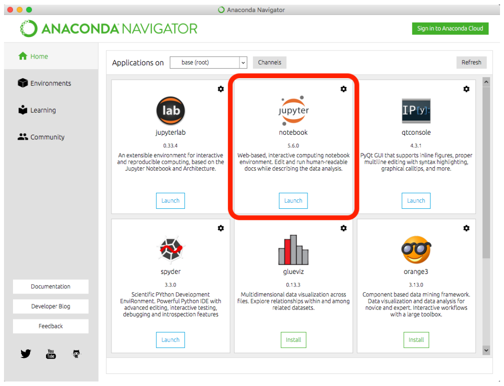
Lab 1 Guide

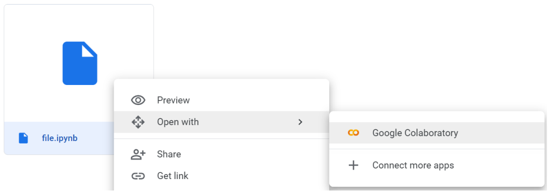
Environment Setup

This quarter, we will be doing all our labs inside Jupyter Notebook files. Some popular ways to work with Jupyter Notebooks (.ipynb files) are:

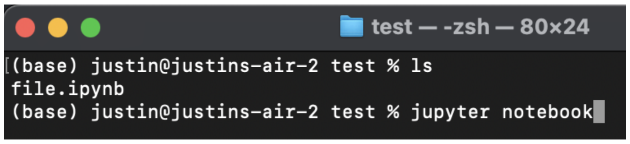
- Anaconda (by using the Anaconda Navigator GUI's Notebook App)



- Google Collaboratory (by opening files directly inside Google Drive)



- Miniconda (by running "jupyter notebook" inside your working directory)



```
test — -zsh — 80×24
[(base) justin@justins-air-2 test % ls
file.ipynb
(base) justin@justins-air-2 test % jupyter notebook
```

Anaconda is more **friendly towards newer coders** with its GUI, and will be the default environment for our labs. However, if you already have experience with the other methods, feel free to use those. Google Collab is a decent option for **notebook sharing**, but can be confusing/tedious to mount/load files locally. Miniconda is a **cleaner and space-saving** version of Anaconda (and also my preferred choice when working with jupyter notebooks).

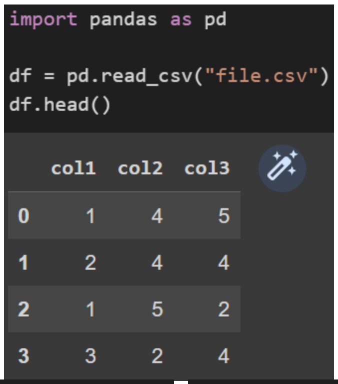
Lastly, Miniconda and Anaconda need to be installed, while Google Collab takes no setup time or disk space.

Libraries

For our labs, we will use the **pandas, numpy, and matplotlib** python libraries extensively, which will mostly deal with **analyzing/interpreting** and **visualizing** data. These libraries are pretty straightforward – most of it is just getting used to the syntax, which comes with practice. I don't like name-dropping a bunch of links to library documentation/explanations, so I've compiled a small list of what I think are the most important/common functions you will use during labs here:

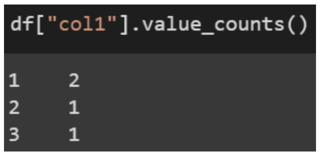
- read_csv() - used for reading in data from a .csv file into a pandas dataframe
- head() - used to view the first few rows of a data frame to get a feel for the data
- loc[: , col_name] - used to grab a specific dataframe column by name (string)
- iloc[: , col_index] - used to grab a specific dataframe column by index (integer)
- value_counts() - used to sum up unique values of one column in a dataframe
- plot() - used to plot the values in a dataframe
- groupby() - used to group up dataframe data by values in a column or columns

Using these functions together might look a little something like this:



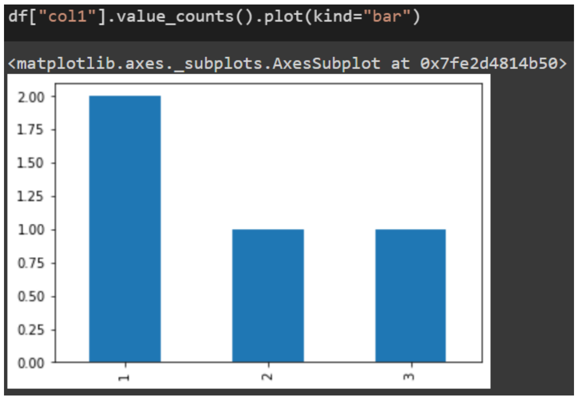
```
import pandas as pd

df = pd.read_csv("file.csv")
df.head()
```

```
   col1  col2  col3
0     1     4     5
1     2     4     4
2     1     5     2
3     3     2     4
```



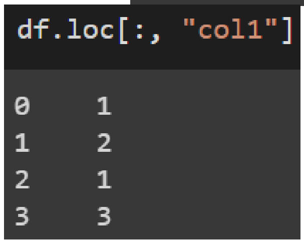
```
df["col1"].value_counts()
```

```
1    2
2    1
3    1
```



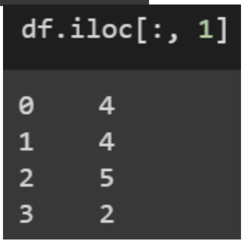
```
df["col1"].value_counts().plot(kind="bar")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2d4814b50>
```



```
df.loc[:, "col1"]
```

```
0    1
1    2
2    1
3    3
```



```
df.iloc[:, 1]
```

```
0    4
1    4
2    5
3    2
```

Lab 1 Tips

Question 0

Give your best guess and explain your thought process behind your answer.

Question 1

A function you may find useful is set_index(*col*).

Question 2

Instead of using a for loop to change all the values of a column, we can use special functions offered by pandas that can **modify entire columns** at a time. For example, if we wanted to **cast** a column of floats (eg. df.col1) as integers, we might use the following line of code:

```
df.col1 = df.col1.astype(int);
```

This will cast the floats at each row in this column as integers. Now let's say we wanted to **apply our own function** to **every row value** (e.g. find the first digit or subtract each integer by 5). Then, we can use the apply() function (along with a function we defined previously, or with a lambda function):

```
df.col1 = df.col1.apply( lambda x: x=x-5 );
```

or

```
def subtract5(x):
    x = x-5;
df.col1 = df.col1.apply(subtract5);
```

Finally, we can visualize a series of numbers using value_counts() and plot() as shown on page 2.

Question 3

Here we do the same thing as Q2, but we are extracting the last digit. An easy way to index the last digit is by using -1 (e.g. 'hello'[-1] returns o).

Question 4

Here we again are doing something similar to Q2. (But which index should we be using?)